

Process Definition in Web-Time (A Fast Start for a Fast-Moving Company)

Abstract:

In small entrepreneurial companies, the most critical things are speed and flexibility. We must be able to move quickly, react to both opportunities and threats, and make new product available to customers on a regular basis. In an environment like this, can we afford to spend the time to focus on developing and improving our software development processes? And can we afford the overhead involved in following well-defined processes?

A better question would be: "Can we afford *not* to address our software development processes?"

This paper will focus on the initial process definition work that I helped a young Internet company to get started on. I will chronicle the steps we took to get a quick start on the process definition they so desperately needed, and how we were able to achieve usable improvements in a relatively short time. We will discuss the challenges we faced and the things we did that helped the project along.

Background

After 13 years at the Software Engineering Institute (SEI) and a year as Manager of Software Process & Quality Assurance at a small company, I had just begun to market my services as an independent consultant in Software Development Processes. With my recent experiences, I was especially well-equipped to help small and growing companies. I had learned a lot about what **not** to do, and even had some familiarity with strategies that work.

The Subject Company is classified as an Internet company. It is a few years old and just made its Initial Public Offering (IPO) last year, after my first contact with them. They were experiencing the growing pains that are common to companies of this type. For example:

- Missed deadlines in spite of heroic effort
- Surprise features and missing features
- Last-minute requirements changes
- Serious problems delivered to test
- Growing animosity among departments

They engaged my services because they knew that their ad-hoc processes were causing them severe problems. And they were smart enough to know that the problems would only get worse as they continued to grow.

Challenge #1: Getting Started

The champion for software process improvement was also the sponsoring executive, and there was no significant resistance to the idea from any level of the company. In fact, everyone seemed excited by the prospect of fixing the processes that were causing so much pain. In spite of this surprising organizational eagerness, getting started turned out to be the most frustrating part of the entire project. My first contact with the company was on 15 October 1999, but we did not set the date for the kick-off activities until 14 January 2000 – a full three months later.

Process Definition in Web-Time

The problem wasn't resistance or lack of interest; rather it was a matter of getting the right people together to discuss, agree and approve the proposed actions. Executives in this sort of company are routinely stretched so thin that redirecting their attention to an important but not urgent (Covey's Quadrant 2) subject take time and persistence. Even the effort's sponsor had to be regularly reminded that she needed to make things happen.

I am not sure that I could have taken actions that would significantly accelerate this part of the project. I believe it is just a cost of doing business with a fast-moving Level 1 organization.

The Kick-off

The process definition effort was finally kicked off on 3 February 2000. The objectives of the kick-off were to:

- Bring everyone to a common understanding of the goals of the process improvement effort,
- Provide a very brief introduction to the CMM (just enough so that everyone could participate in the rest of the kick-off activities),
- Generate discussion of the problems that the various parties were experiencing in their current software development processes (so I could understand the biggest issues they were facing at the time), and
- Administer the SEI assessment questionnaire (which would provide a basis for the initial process improvement strategy).

The company was split into two parts for the kick-off event:

- In the morning, the engineering staff (including Tech Writers and QA) participated, and
- In the afternoon, the management and marketing folks went through it.

This was done for several reasons:

- To make the best use of each person's time (this is a significant issue for a small fast-moving company),
- To allow the presentation and discussions to be focused differently for the two audiences (engineers were more interested in how the CMM would affect their work, and managers were more interested in the business case for the project), and
- To assure that everyone would feel free to speak openly about the problems they perceived in their current practices.

The kick-off event was successful in all respects. The only problem we experienced was that the managers and marketing folks required almost three weeks of badgering before they all had submitted their assessment questionnaires. Naturally, this is part of the problem discussed under "Challenge #1", above.

Challenge #2: Interpreting the Mini-Assessment

In a full CMM-Based Assessment for Internal Process Improvement (CBA-IPI), the assessment questionnaire is not the main source of information, rather it is used to focus the discussion during interviews and other assessment activities on potential issues. In a Mini-Assessment, the questionnaire (although it is supplemented with information gleaned from various discussions) becomes the main source of information about the company's process maturity and issues. The challenge in a Mini-Assessment is in making sense of the results of the questionnaire.

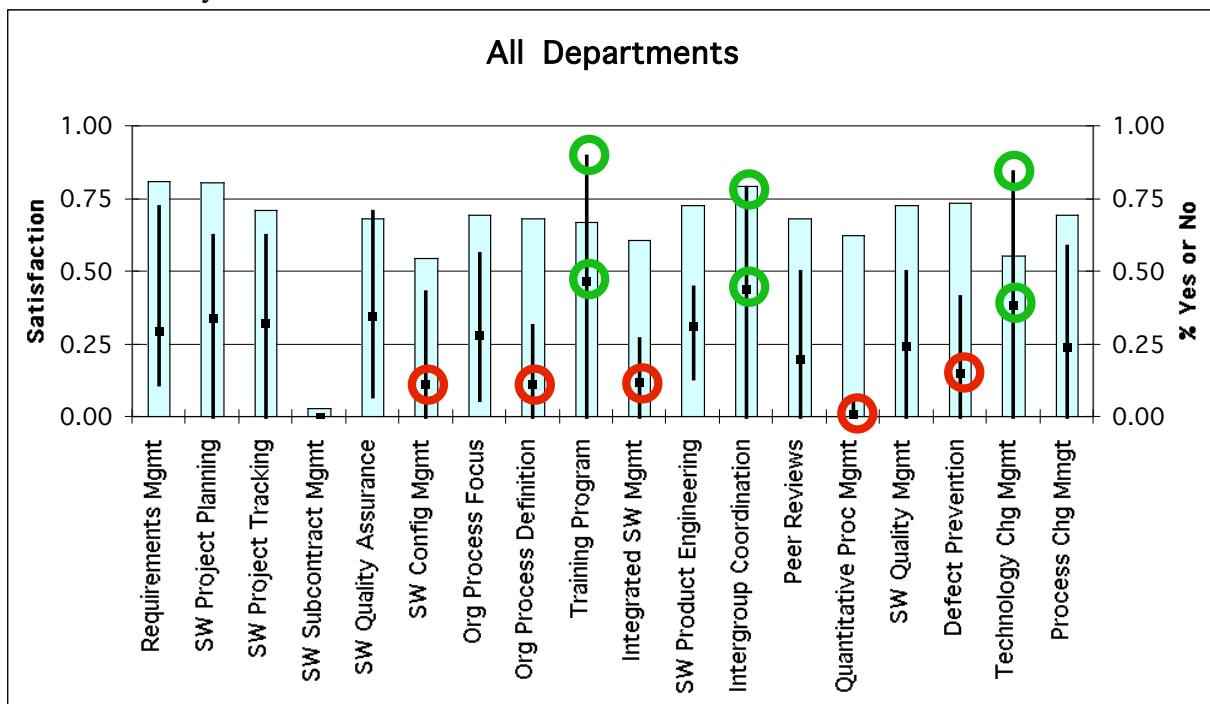
Process Definition in Web-Time

The questionnaire is organized by Key Process Area (KPA), and each question is roughly related to one of the goals for a KPA. For each question, the responder is asked to check “Yes” (the goal is satisfied), “No” (the goal is not satisfied or only partly satisfied), “Does Not Apply” (the goal does not apply to this organization) or “I Don’t Know”, and space is available for written comments.

For any particular question, some people answered “Yes”, others answered “No” and others checked another answer or left it blank. Given this variety of opinion and lack of options for verifying them, how can we come up with a coherent picture of the organization’s process maturity from this data? I chose to use a two-dimensional view for each question:

- How highly they rated themselves – What percentage of the people who answered “Yes” or “No” answered “Yes”? ($\#Yes / (\#Yes + \#No)$) This gives a sense of how likely it may be that the organization actually satisfied the goal.
- How strongly they hold that view – What percentage of the people answered “Yes” or “No”? ($(\#Yes + \#No) / \#Participants$) This tells us what portion of the organization felt they could answer the question.

For each KPA, I combined the results of the individual questions for a single composite view: That analysis resulted in this chart:



- Each black line with a square on it gives a composite view of the “How highly they rated themselves “scores for a KPA. It is a range-plot that shows the highest, lowest and mean ratings of the questions for that KPA.
- Each blue bar gives a composite view of the “How strongly they hold that view” scores for a KPA. It shows the mean of the strengths for the questions for that KPA.
- The Green circles highlight the three highest-rated KPA’s, and the red circles highlight the four lowest-rated KPA’s.

Process Definition in Web-Time

For an example of reading this chart, refer to the first column, "Requirements Mgmt":

- The top of the black line is at almost 75%, meaning that on the best question for this KPA, nearly 75% of the people who answered "Yes" or "No" answered "Yes".
- The bottom of the black line is at about 12%, meaning that on the worst question for this KPA, only about 12% of the people who answered "Yes" or "No" answered "Yes".
- The black square on the line is at about 30%, meaning that on average, only about 30% of the people who answered "Yes" or "No" to questions for this KPA answered "Yes".
- The blue bar goes up to about 80%, indicating a strong opinion with an average of 80% of the participants answering "Yes" or "No" to the questions for this KPA (as opposed to "I Don't Know", "Doesn't Apply" or blank).

The written comments provided important information about people's misunderstandings about the terminology used or the instructions for the questionnaire. The net result of these misunderstandings is that most of the ratings should be lower than shown on the chart. For example:

- "Yes – but..." – Quite often, people checked "Yes", but then explained in the comments how those things were not done regularly or under certain conditions. The instructions clearly state that in these cases you should respond "No – but..."
- "Does the project follow a written policy/procedure/standard..." – Most of the KPA's include a question of this type. These questions have three distinct foci:
 - Does the policy/procedure/standard exist,
 - Is it written down, and
 - Is it followed consistently?

From my discussions with people during the kick-off and at other times, I know that none of their policies is written down, so all of those questions should have been answered "No".

- "Are measurements used to determine the status..." – Most of the KPA's include a question of this type. The answers to these questions should all have been "No" because:
 - The company has no measurement program or metrics database, and I saw no evidence that any measurements are taken for anything, and
 - The comments indicated that most people interpreted the questions to be asking about measurements of work products, when they are actually referring to measurements of the processes themselves.
- "Are the activities for [the process] subjected to SQA [Software Quality Assurance] audit or review?" Again, the comments indicated that most people interpreted the questions to be asking about audits or reviews of work products, when they are actually referring to audits or reviews of the processes themselves.

Mini-Assessment Results

I published the results of the Mini-Assessment on 2 March 2000, exactly a month after the Kick-off. (The delay was mainly due to waiting for surveys from the Marketing and management folks.) The results of the Mini-Assessment were not unexpected: the company was clearly Level 1. This in spite of the fact that most of the ratings should have been lower because of people answering "Yes" when they should have said "No" (see "Challenge #2", above). The consensus was that no goal for any Key Process Area (KPA) was completely satisfied, and many goals were not even thought about.

Process Definition in Web-Time

One of the few bright spots from the Mini-Assessment was that they rated themselves surprisingly high on the Inter-Group Coordination KPA. They chose to capitalize on this perceived strength, making it a rallying cry: “Let’s continue to work together as we solve the process problems we have identified.”

Besides the “All Departments” analysis represented by the chart above, I also analyzed the data for each department to try to identify serious discrepancies in their views. I found that there was general agreement about most of the KPA’s across all of the departments. Only two KPA’s showed any significant disparity, and one of them was due to a consistent misinterpretation of the term “Defect Prevention” among the development staff. That left only one KPA with true disagreement; but since it was not a Level 2 KPA, this disagreement was not addressed in the initial planning.

Although the Mini-Assessment provided no surprises, and mostly confirmed what they already knew, there was none-the-less significant value in the exercise because it:

- Introduced a new vocabulary that they immediately began to use to talk about their process problems,
- Was a shared experience – it was not a matter of one person pointing a finger at anyone else; rather the whole company was point a finger at itself,
- Highlighted senior management’s commitment to the process improvement work – committing everyone’s time to the kick-off event made it clear that it is very important to the company’s future, and
- Established momentum for the harder work that would follow.

Challenge #3: Identifying the Initial Priorities

We set the initial strategy using the results from the Mini-Assessment and these principles:

- CMM Principle: Focus on Level 2 first,
- Attack things that everyone agrees are important problems, and
- Look for “low-hanging fruit” to generate some early wins.

The Software Subcontract Management KPA was a non-issue because this company has no subcontractors.

Software Configuration Management (SCM) was the lowest-rated of the other Level 2 KPA’s, suggesting that it should be an early priority. However it was not chosen for priority action because:

- We saw little opportunity for quick improvements (they already used automated code control, and most other SCM issues take significant time and effort to work out), and
- Many people did not see SCM as a problem area. (This would change, later!)

The remaining four Level 2 KPA’s were equally weak, generally recognized as problem areas, and each provided obvious opportunities for quick wins. So, three process teams were formed to address these four KPA’s.

Requirements Management

Everyone highlighted the Requirements Management KPA as a significant pain-point in the organization. No one was happy with the way it worked, and everyone could see opportunities for easy changes that would yield fast returns (though there was not agreement on exactly what

Process Definition in Web-Time

those changes should be). Requirements Management was also recognized as an important basis upon which all other work was dependent; so everyone agreed that it should be the top priority.

The Requirements Management team's initial goals were to:

- Establish the requirements definition process and
- Refine the requirements template (agreeing on its contents and definitions of terms).

This team did **not** attempt to address the requirements change process, though everyone agrees that it would be a logical follow-on to the initial work.

Project Management

The Software Project Planning and Software Project Tracking & Oversight KPA's were the other significant pain-points in the organization. There was significant confusion about how projects were initiated and planned, and problems with understanding the current status of projects was a recurring theme. There were clearly ample opportunities for quick improvements that would be quite beneficial. And like Requirements Management, Project Management was seen as a basic activity upon which all projects depend.

The Project Management team's initial goals were to:

- Define the terms that are used in project planning and tracking (e.g. do estimates of engineering time include meetings, interruptions and incidental activities?),
- List all of the activities that must be planned and tracked for a project (including those done by QA, the technical writers and Marketing), and
- Define the process for creating a project plan.

Again, note that the initial goals do not include managing changes to plans.

Software Quality Assurance

SQA was not identified as a particular problem area, but we decided to include it in the initial priorities because it comprised significant opportunities. The SQA function was just being built from scratch, and everyone wanted to get it started off on the right foot.

The Quality Assurance Team's initial goals were to:

- Define QA's role in other departments' activities (e.g. requirements definition, project planning, design reviews),
- Develop standards for test planning
- Define a standard testing process

Notice that the initial goals did not include any process assurance activities, only product assurance. This was done because product assurance had already begun to cause problems, and also because there were not yet any formal processes to assure. But with this in mind, the organizational structure and philosophies are being formed so that process assurance can be added as it becomes reasonable to do so.

Challenge #4: Establishing a Common View

The three process improvement teams held their first meeting on 5 April 2000, a month after the Mini-Assessment findings were published. As with getting started, this delay was mainly due to the difficulty of getting senior management to discuss and agree on the actions to be taken.

Process Definition in Web-Time

In the very first set of meetings for the three teams, it became apparent that there was no common view of their software development process. Different people listed different sets of activities, and used the same words to mean different things. The disparity was most obvious between departments, though even within the development department there were significant differences among individuals.

By the end of the second meetings, a new highest-priority goal had been identified for the teams: Work with the other two teams to agree on a single description of the software development process. This included listing the steps in the process and defining the terms that were used, as well as identifying the parts of the over-all process on which each team would be focusing.

The project management team became the focal point for this effort, postponing work on their initial goals for the time being. The other two teams continued working toward their initial goals while participating in this work. The first one-to-two months of team meetings were spent agreeing on this common view of the development process. It took much more effort than anyone had anticipated, but it was a very educational and valuable exercise, and it provided the needed basis upon which the other work could be built.

In retrospect, it would have been good to collapse the three teams into one when we identified the need for a common view of the development process. It likely would have facilitated the definition work and allowed all three teams to refocus back on their initial goals more quickly.

Postmortem Before Starting

In late June 2000, as the three teams were working toward their initial goals, the company completed the development project that they had been working on. This provided a unique opportunity to do a postmortem analysis of the project to provide additional input to their process development teams.

This postmortem analysis yielded two important insights:

- Software Configuration Management (SCM) is a bigger problem than they realized. They discovered that they need to institute change control on all types of work products in order to bring some sanity to their work. This finding confirmed the results of the Mini-Assessment, and so SCM will be attacked next.
- Inter-Group Coordination is a root cause of many other ills. This contradicted their earlier opinion, showing that their ability to work together is not as good as they thought it was. They realized that although there are relatively good relationships among the groups, they need some formal mechanisms to insure that all coordination takes places as needed.

Initial Changes in Pilot Test

With a new development project kicking off in July 2000, the three teams focused on identifying specific process changes that they could pilot test on the new project. Because of their early focus on “low-hanging fruit”, they already had several process changes ready for pilot use.

Process Definition in Web-Time

These changes are currently being pilot tested:

- Requirements Management:
 - The formal Requirements Definition process includes steps for proposing, evaluating and prioritizing requirements, and for deciding on the actual content for the product version.
 - The Requirements Template includes content guidelines for all sections and is based on commonly accepted definitions of terms.
- Project Management
 - The new Engineering templates provide an intermediate view of the system between the Requirements and Design specifications. This intermediate view is designed to facilitate the Engineering staff's evaluation of proposed Requirements, allow them to make more reasonable effort and schedule estimates, and provide a way to validate that the Design that is eventually specified accurately represents the intent of the Requirements Specification.
 - Added structure within the Engineering department is designed to allow them to more effectively carry out their wide variety of concurrent activities (e.g. requirements analysis for future versions, design & development for the current version, maintenance of the past version)
- Software Quality Assurance
 - Active early involvement of QA (and the technical writers) during Requirements Analysis and Design Review activities is supposed to improve the quality of the Requirements and Design Specifications, and at the same time, give the Quality Engineers a better understanding of the product that they will validate.
 - The new Test Planning process and standards should assure that reasonable, but complete tests have been specified and prepared while the software is in development.
 - The Testing process assures that both testing and problem tracking do not allow problems to "fall through the cracks".

The initial experience with these process changes has been positive, and everyone is enthusiastically looking forward to the postmortem analysis of this project.

Conclusion

This company has done a commendable job of getting a fast start on their process definition work. From the date of their Mini-Assessment to the beginning of pilot testing some significant process changes was only 5 months. This is much faster than many process improvement efforts can move.

These process improvements should dramatically improve the stability of their current project, and demonstrate the value of process improvement. They should also provide the momentum that the company will need to continue with their process improvement work, especially the difficult job of establishing the change control mechanisms that they now recognize they need.

At the same time, it should be noted that these steps are only the beginning of a long process improvement effort. By themselves, these steps do not even bring the company close to achieving CMM Level2. Like any other company in any industry, process improvement for an Internet company is a long-term effort, even if it begins with a few simple steps.

Process Definition in Web-Time

Bio:

Alan S. Koch's 23 years in software development includes:

- 9+ years designing, developing and maintaining software,
- 5+ years in Quality Assurance (including establishing & managing a QA department),
and
- 2+ years in Software Process Improvement.

Mr. Koch was with the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU) for 13 years where he became familiar with the Capability Maturity Model (CMM), earned the authorization to teach the Personal Software Process (PSP) and worked with Watts Humphrey in pilot testing the Team Software Process (TSP).

After leaving the SEI, Mr. Koch was the Manager of Software Process and Quality Assurance for a software company. He is now an independent consultant helping companies to improve the return on their software investment by focusing on the quality of both their software products and the processes they use to development them.